Department of Informatics
King's College London
United Kingdom

7CCSMPRJ Individual Project

A Framework for Evaluating Speech-Based ML Models for Remote Health Assessment

---

Name: **Ayush Pradhan**
Student Number: [redacted]
Course: MSc Data Science

Supervisor: Dr. Nicholas Cummins & Dr. Sophia Tsoka

This dissertation is submitted for the degree of MSc in MSc Data Science.

Abstract

The application of machine learning to speech data offers a promising, scalable avenue for the remote assessment of health conditions, particularly Major Depressive Disorder (MDD). However, the translation of these technologies into reliable clinical tools is often hampered by methodological inconsistencies and a lack of robustness in evaluation. This dissertation develops and applies a comprehensive evaluation framework to critically investigate the impact of feature representation and cross-validation strategy on the performance and stability of depression detection models.

This project provides a systematic, empirical comparison of three distinct feature sets: a curated handcrafted acoustic set (MSHDS), a comprehensive handcrafted set (OpenSMILE), and learned embeddings from a foundational model (Wav2Vec2). These features are evaluated on the clinically relevant Androids Corpus, which contains both read (scripted) and spontaneous (interview) speech from individuals with depression and healthy controls. The framework assesses performance using two classifiers, a Support Vector Machine (SVM) on summary statistics and a deep learning-based CNN-LSTM on full sequences, across both a standard k-fold and a more robust nested k-fold cross-validation protocol.

The results reveal a complex interaction between feature type, speech task, and evaluation methodology. Spontaneous speech was found to be significantly more informative than read speech, particularly for the high-dimensional OpenSMILE feature set. The nested cross-validation protocol consistently produced more conservative but more stable performance estimates, highlighting the optimistic bias of simpler methods. Critically, the CNN-LSTM model, when trained on sequential Wav2Vec2 embeddings from spontaneous speech and tuned with Bayesian optimization (Optuna), achieved the highest discriminative power (Mean AUC: 0.865). This work demonstrates the necessity of rigorous, multi-faceted evaluation and provides a methodological blueprint for developing more trustworthy speech-based health assessment tools.

Contents

List of Figures

List of Tables

1 Introduction

This dissertation project is situated at the intersection of digital health, speech signal processing, and machine learning, with a specific focus on applications in mental health. It provides the background and context of the work.

1.1 Background and Rationale

The proliferation of mobile and smart devices has driven a fundamental transformation in healthcare, enabling the continuous and remote monitoring of individuals' health states. This field, broadly termed digital health, offers the potential to move from reactive, episodic clinical assessments to proactive, preventative models of care [1]. Within this domain, speech analysis has emerged as a uniquely powerful modality. The human voice is a rich, non-invasive, and readily accessible data stream that carries a wealth of information beyond its linguistic content. Acoustic and prosodic characteristics of speech are modulated by an individual's physiological, cognitive, and affective state, making the voice a potent biomarker for a range of health conditions, particularly in neurology and mental health [2].

Major Depressive Disorder (MDD) represents a significant global health challenge, affecting an estimated 280 million people worldwide and standing as a leading cause of disability [3]. Traditional assessment relies on clinical interviews and patient-completed questionnaires, which, while essential, are subjective and conducted infrequently. Speech-based analysis presents an opportunity to augment this process with objective, high-frequency data, potentially enabling earlier detection of depressive episodes and more timely intervention [4]. However, for these technologies to be integrated responsibly and effectively into clinical practice, the machine learning models that underpin them must be robust, reliable, and well-understood [5].

The central challenge, and the primary motivation for this dissertation, is that the reported performance of speech-based models can be highly sensitive to methodological choices. Variations arising from recording equipment, acoustic environments, and natural intra-speaker vocal changes introduce significant variability into the data [6]. Furthermore, while the choice of feature representation and model architecture are significant, it is the evaluation strategy that most critically influences a model's perceived effectiveness. A failure to employ rigorous validation techniques can lead to over-optimistic performance estimates and models that do not generalize to new, unseen data, hindering clinical translation and eroding trust in the technology [7]. This project addresses this critical gap by focusing on the development of a framework to systematically evaluate and compare the robustness of different speech analysis pipelines.

1.2 Aims and Objectives

The overarching aim of this project is to develop and apply a methodological framework to evaluate and compare the efficacy of various speech-based machine learning

techniques, assessing their suitability for the robust detection of depression in real-world health research applications.

This aim is guided by the following primary research question:

To what extent does the choice of cross-validation methodology (e.g., standard vs. nested k-fold) influence the comparative evaluation of different acoustic feature representations in terms of predictive performance and feature selection stability for depression detection?

To address this question, the project is structured around the following specific objectives:

1. To design and implement a modular Python-based framework for conducting systematic machine learning experiments on speech data.
2. To implement and compare three distinct feature extraction approaches: a curated handcrafted set (MSHDS), a large-scale comprehensive set (OpenSMILE), and modern learned embeddings (Wav2Vec2).
3. To evaluate these features using two classifiers: a classical Support Vector Machine (SVM) and a deep learning-based CNN-LSTM.
4. To systematically compare two cross-validation strategies: a standard k-fold and a robust nested k-fold with Bayesian hyperparameter optimization.
5. To analyze the impact of speech elicitation type (read vs. spontaneous speech).
6. To critically evaluate the results based on performance and stability to draw conclusions about best practices for robust evaluation.

The complete source code, experimental notebooks, and instructions for reproducing these results are submitted as supplementary material and are also available in the project's version-controlled repository on GitHub: Speech-Analysis-Framework.

1.3 Dissertation Structure

This dissertation is structured as follows:

- **Section 2:** provides a systematic review of the relevant background.
- **Section 3:** details the design of the experimental framework, the dataset, and the implementation of the pipelines. This chapter also presents and critically analyzes the findings from the 24 core experiments.
- **Section 4:** interprets the results, discusses limitations, and provides a concluding summary of the project's contributions and suggestions for future work.
- **Section 5:** discusses the legal, social, ethical, and professional issues related to the project.

2 Background Theories and Literature Review

This chapter provides a systematic review of the technical background underpinning the project. It begins by establishing speech as a digital biomarker for health, then examines the core challenges of model reliability that motivate this research. Subsequently, it critically compares the dominant approaches to feature representation in speech analysis and examines the theoretical foundations of rigorous machine learning evaluation methodologies. Finally, it discusses the crucial influence of the speech elicitation task on model performance.

2.1 Speech as a Digital Biomarker for Health

The field of digital health has seen a surge in the exploration of novel biomarkers capable of being captured remotely and unobtrusively. Among these, speech holds a privileged position due to its high information density and the prevalence of recording devices [8]. A digital biomarker is defined as an objective, quantifiable physiological or behavioral measure collected and measured by means of digital devices [9]. Speech acoustics and prosody, as direct products of the complex neuromuscular actions of the respiratory, laryngeal, and articulatory systems, are modulated by both central and peripheral nervous system function, making them sensitive indicators of health status [2].

Research has demonstrated the utility of speech analysis across a wide range of conditions. In neurodegenerative diseases such as Parkinson's Disease and Amyotrophic Lateral Sclerosis (ALS), vocal biomarkers like articulatory precision, phonation stability (jitter and shimmer), and speech rate have proven effective for tracking disease progression [10, 11]. In the domain of mental health, the link between affective state and vocal expression is well-established. Psychomotor retardation in depression, for example, often manifests as slowed speech, longer and more frequent pauses, and a reduction in pitch variability (monotone speech), while states of anxiety or agitation can correspond to a higher fundamental frequency and increased speech rate [12]. Large-scale projects like the RADAR-MDD study have incorporated remote speech collection via smartphones as a core component in their longitudinal monitoring of Major Depressive Disorder, highlighting its perceived value in capturing dynamic symptom changes [4].

However, as emphasized by Dineley et al. [5], for speech analytics to successfully transition from promising research to clinically-accepted tools, a shift in perspective is required. It is not enough to generate "features"; the field must develop psychometrically sound "measures" that are reliable, valid, and interpretable. This necessitates a deep understanding of the sources of variability and a commitment to rigorous methodological validation, which forms the central theme of this dissertation.

2.2 Feature Representation in Speech Analysis

The process of converting a raw audio waveform into a set of numerical inputs for a machine learning model is known as feature extraction. The choice of feature representation is a critical design decision that reflects a particular assumption about what information

in the speech signal is most important, the literature reveals three dominant approaches to this end.

2.3 Handcrafted Feature Sets

Historically, speech analysis has relied on handcrafted, or engineered, features. These are algorithms designed by experts based on established knowledge from phonetics, acoustics, and signal processing. They can be broadly categorized into two groups.

2.3.1 Comprehensive "Brute-Force" Sets

This approach is predicated on the idea of extracting a vast, comprehensive set of acoustic parameters and allowing the machine learning model to determine which are most relevant. The openSMILE toolkit is the exemplar of this philosophy [13]. Standardised feature sets like the Interspeech 2016 Computational Paralinguistics Challenge (ComParE) set provide over 6,000 features, covering an extensive range of acoustic domains including prosody, voice quality, and spectral characteristics [14]. While this approach ensures broad coverage, it often results in highly redundant, high-dimensional feature spaces that are prone to overfitting and can be computationally expensive to work with [7].

2.3.2 Curated, Expert-Driven Sets

A more recent trend advocates for the use of smaller, curated feature sets based on theoretical and clinical relevance. The concept of a Minimal Speech Health Data Set (MSHDS), for example, prioritizes features with strong psychometric properties and clear links to the underlying speech production mechanism [5]. This approach often involves selecting robust measures like mean fundamental frequency (F0), phonation rate, and Cepstral Peak Prominence (CPP), a reliable indicator of voice periodicity [15]. MSHDS explicitly excludes less reliable features like jitter and shimmer, which can be highly sensitive to recording conditions and algorithmic implementation [16]. This expert-driven approach yields a more interpretable and computationally efficient feature set, but risks overlooking complex, non-linear interactions that a larger set might capture.

2.4 Learned Feature Representations: The Rise of Foundation Models

The third paradigm avoids manual feature engineering entirely. Instead, it leverages deep learning to learn optimal feature representations directly from data. This is the core principle of transfer learning and foundation models. Large-scale models, often based on the Transformer architecture, are pre-trained on thousands of hours of unlabelled speech data in a self-supervised manner.

Models like Wav2Vec2 [17] learn rich, contextualized representations of the speech signal by solving a contrastive task: identifying the true speech segment from a set of distractors. The resulting embeddings, extracted from the model's hidden layers, serve as a powerful, general-purpose feature set. Studies have consistently shown that these learned

features often outperform traditional handcrafted sets in paralinguistic tasks, including depression detection [18]. Their strength lies in their ability to capture complex, hierarchical patterns in the data that are difficult to define with explicit algorithms, and their robustness, which is learned from exposure to highly diverse data during pre-training.

2.5 Methodological Rigor in Model Evaluation

The credibility of any machine learning model hinges on the rigor of its evaluation. In clinical applications, where models may inform decisions with significant consequences, this rigor is paramount. The literature highlights two critical aspects of evaluation: the cross-validation strategy and the stability of the model's findings.

2.6 Standard vs. Nested Cross-Validation

K-fold cross-validation is a standard technique for estimating a model's generalization performance. However, a common methodological flaw arises when hyperparameter tuning (such as selecting the optimal number of features or tuning a model's learning rate) is performed. If the same data used to evaluate the final model is also used to inform these tuning decisions, information "leaks" from the test set, leading to an optimistically biased performance estimate [19].

As demonstrated theoretically and through simulation by Ghasemzadeh et al. [7] and Vabalas et al. [20], this bias can be substantial, leading to the publication of results that are not reproducible on new data. The correct approach is **nested cross-validation**. This method uses an "outer loop" to split the data for final performance evaluation and a separate, independent "inner loop" to perform hyperparameter tuning using only the training portion of each outer fold. This strict separation ensures that the final performance estimate is unbiased, providing a more realistic expectation of how the modeling pipeline would perform in a real-world scenario. A key aim of this dissertation is to empirically investigate the magnitude of this optimistic bias on a real clinical dataset.

2.7 Feature and Model Stability

Beyond predictive accuracy, the reliability of a clinical model depends on its stability. **Feature stability** refers to the consistency with which a feature selection algorithm identifies the same features as important across different subsets of the data [21]. A model that relies on a completely different set of features for each fold of cross-validation is not trustworthy, even if its average accuracy is high. This suggests it may be learning spurious correlations rather than robust underlying patterns. Similarly, the performance of a model itself should be stable. A model that achieves 95% accuracy on one fold and 55% on another is less reliable than a model that consistently achieves 75% across all folds.

2.8 The Influence of Speech Elicitation Task

A final, crucial consideration in speech-based health assessment is the nature of the elicitation task. The acoustic properties of speech can vary significantly depending on whether the speech is scripted and read, or spontaneous and conversational [22].

- **Read Speech:** Tasks like reading a phonetically-balanced passage (e.g., "The North Wind and the Sun") provide a controlled sample of a person's voice, minimizing linguistic variability. However, this type of speech is often produced with an unnaturally flat prosodic contour and may not be representative of a person's typical affective state [23].
- **Spontaneous Speech:** Elicited through semi-structured interviews, spontaneous speech is far richer in prosodic and emotional variation. It contains natural hesitations, changes in pace, and pitch contours that are highly indicative of affective state [24]. However, it is also much more variable and "noisy".

The choice of feature set may interact with the speech task. A key part of this dissertation's analysis will be to investigate this interaction.

3 Methodology, Implementation, and Results

This chapter details the technical contributions of the project. It outlines the design of the experimental framework, describes the dataset, details the feature extraction and modeling pipelines, and finally, presents and critically analyzes the results from the comprehensive suite of experiments.

3.1 The Evaluation Framework

The primary technical contribution of this project is a modular, Python-based evaluation framework designed to systematically investigate the research questions. The framework was built to be reusable, with a clear separation of concerns between data handling (`src/data_loader.py`), feature extraction (`src/mshds_extractor.py`, `src/opensmile_extractor.py`, `src/foundation_model_extractor.py`), and model evaluation (`src/cv_strategies.py`, `src/dl_cv_strategies.py`). All code was developed in Python 3.11 within a fully containerized Conda environment (`msc_final_env`) to ensure complete reproducibility. Key libraries include Pandas, scikit-learn, PyTorch, Parselmouth, and Optuna. The entire project is version-controlled using Git with a feature-branching workflow, and the codebase is submitted as a supplementary artefact.

3.2 Dataset: The Androids Corpus

The dataset chosen for this investigation is the Androids Corpus, a publicly available, clinically relevant corpus designed for speech-based depression detection [25]. It was selected for several key reasons:

- **Clinical Ground Truth:** The corpus contains speech from 59 individuals with a clinical diagnosis of MDD, confirmed by psychiatrists, and 59 healthy controls, providing a reliable binary classification label.
- **Participant Matching:** The patient and control groups are carefully matched for age, gender, and education level, which helps to minimize the influence of these potential confounding variables.
- **Dual Elicitation Tasks:** The inclusion of both a scripted **Reading Task** ("The North Wind and the Sun") and a spontaneous, semi-structured **Interview Task** allows for a direct comparison of these two distinct speech types.
- **Real-World Conditions:** Data was collected in mental health centers using standard laptop microphones, reflecting more realistic "in-the-wild" conditions than pristine laboratory recordings.

The dataset includes pre-defined 5-fold splits for cross-validation, which were used in this study to ensure comparability. The raw data consists of 112 reading task audio files and 116 interview sessions, with the interviews pre-segmented into 866 participant-only audio clips. After filtering for very short, non-speech clips, a total of 111 reading files and

857 interview clips were used in the feature extraction pipelines. The class distribution of the 111 unique participants was found to be well-balanced, with 57 patients (51.4%) and 54 controls (48.6%).

3.3 Methodology and Implementation

The pipeline is structured into feature extraction, model training, and stability analysis. Three feature sets (MSHDS, OpenSMILE, Wav2Vec2) were evaluated using two classifiers (SVM, CNN-LSTM) and two cross-validation strategies (Standard K-Fold and Nested K-Fold with Optuna tuning). For the Interview Task, clip-level features were aggregated to the session level by calculating their mean and standard deviation.

3.3.1 Feature Extraction Pipelines

Three distinct feature extraction pipelines were implemented, each corresponding to a different feature philosophy. For the Interview Task, clip-level features were first extracted and then aggregated to the session level by calculating their mean and standard deviation, resulting in a single feature vector per participant for each task.

1. **MSHDS (Curated Handcrafted):** A set of 25 acoustic features inspired by the Minimal Speech Health Data Set was implemented in `src/mshds_extractor.py` using the Parselmouth library. This set focused on robust measures of phonation (Mean F0, HNR, CPP), timing (Speaking Rate, Pause Rate), intensity, formants, and spectral moments.

2. **OpenSMILE (Comprehensive Handcrafted):** The OpenSMILE toolkit was used to extract a large-scale feature set based on a modified Interspeech 2009 configuration provided with the corpus (`Androids.conf`). This resulted in a 911-dimensional feature vector for each audio sample, covering a wide range of acoustic domains.

3. **Wav2Vec2 (Learned):** The pre-trained foundational model `facebook/wav2vec2-base-960h` was used as a feature extractor, implemented in `src/foundation_model_extractor.py`. For each audio file, the full sequence of embeddings from the model's final hidden layer was extracted. For the SVM experiments, these sequences were mean-pooled to produce a 768-dimensional summary vector. For the CNN-LSTM, the full, unpooled sequences were used.

3.3.2 Feature and Sequence Aggregation

A critical step for the Interview Task was to aggregate the features from multiple variable-length clips, one clip for each participant's reply to a question, into a single representation for each participant. The aggregation strategy was tailored to the classifier architecture.

For the SVM experiments, which operate on static feature vectors, clip-level summary features were aggregated by calculating their mean and standard deviation. This

produced a single feature vector per participant that captured both the central tendency and the variability of their speech.

For the CNN-LSTM, which requires a single, continuous sequence, a temporal aggregation strategy was implemented. As shown in Listing 1, the individual embedding sequences from each of a participant's clips were concatenated end-to-end to form one long, session-level sequence. This approach preserves the crucial temporal dynamics for the model to analyze. Implemented in (`src/utils.py`).

```python
def aggregate_interview_sequences(clip_sequences, interview_metadata_df):
    """
    Concatenate individual clip sequences into one long sequence per participant.
    """
    # Group all clip filenames by their unique participant ID.
    participant_clips = interview_metadata_df.groupby('unique_participant_id')['filename'].apply(list)

    session_sequences = {}
    for participant_id, clip_filenames in participant_clips.items():

        # Collect all sequence arrays for the current participant.
        participant_sequences = [
            clip_sequences[fname] for fname in clip_filenames
            if fname in clip_sequences
        ]

        # If any valid sequences were found, stack them vertically.
        if participant_sequences:
            session_sequences[participant_id] = np.vstack(participant_sequences)

    return session_sequences
```

Listing 1: Python function for aggregating individual clip sequences into a single session-level sequence via concatenation.

3.4 Classifier Architectures and Evaluation

Two classifiers, a linear Support Vector Machine (SVM) and a deep learning-based CNN-LSTM, were evaluated across all feature sets and data types. A robust nested cross-validation (5x3) protocol was used for all final performance reports. The inner loop performed hyperparameter tuning, the feature selection for the SVM and a 25-trial Bayesian optimization search (Optuna) for the CNN-LSTM. This was compared against a simpler standard k-fold evaluation to assess methodological bias.

3.4.1 Support Vector Machine(SVM)

A linear Support Vector Machine (`sklearn.svm.SVC`) was used as the classical baseline classifier. It was applied to the session-level summary feature vectors (MSHDS, OpenS-

MILE, and mean-pooled Wav2Vec2). The primary hyperparameter tuned for this model was the number of features selected via `SelectKBest`.

3.4.2 CNN-LSTM with Attention

For processing the full, unpooled Wav2Vec2 sequences, a deep learning model was designed and implemented in PyTorch (`src/models.py`). The architecture, shown in Listing 2, is a Convolutional Neural Network-Long Short-Term Memory (CNN-LSTM) network. This design was chosen to leverage the strengths of both architectures:

- **1D Convolutional Layers** (implemented as residual blocks) are used first to act as learnable filters, extracting local, time-invariant patterns from the input sequence.
- **A Bidirectional LSTM** layer then processes the output of the CNNs, allowing it to model long-range temporal dependencies and contextual information from both past and future time steps.
- **A Self-Attention Pooling** layer aggregates the LSTM outputs by learning to assign more weight to the most informative time steps, creating a final, context-aware representation for classification.

```
class CNNLSTM(nn.Module):
    def __init__(self,
                 input_dim=768,
                 num_classes=2,
                 cnn_out_channels=128,
                 lstm_hidden_dim=128,
                 dropout_rate=0.5,
                 activation_fn='silu'):

        super(CNNLSTM, self).__init__()

        # Use residual blocks for stable CNN feature extraction
        self.res_block1 = ResidualBlock(input_dim, cnn_out_channels,
    activation_fn=activation_fn)
        self.res_block2 = ResidualBlock(cnn_out_channels,
    cnn_out_channels, activation_fn=activation_fn)

        # Bidirectional LSTM to capture temporal context
        self.lstm = nn.LSTM(
            input_size=cnn_out_channels,
            hidden_size=lstm_hidden_dim,
            num_layers=2,
            batch_first=True,
            bidirectional=True,
            dropout=dropout_rate
        )


```

```
        # Attention layer to pool sequence information intelligently
        self.attention_pooling = AttentionPooling(input_dim=
    lstm_hidden_dim * 2)

        # Final classifier head
        self.dropout = nn.Dropout(dropout_rate)
        self.fc = nn.Linear(lstm_hidden_dim * 2, num_classes)
```

Listing 2: The `__init__` method of the CNN-LSTM model, defining its architecture.

3.5 Cross-Validation Strategies and Hyperparameter Tuning

A robust nested cross-validation (5x3) protocol was used for all final performance reports to prevent optimistic bias from information leakage.

3.5.1 SVM Tuning via Grid Search

Two CV strategies were implemented in `src/cv_strategies.py` and `src/dl_cv_strategies.py`.

1. **Standard K-Fold:** A 5-fold stratified cross-validation. For the SVM, a fixed number of 25 features were selected using `SelectKBest`. For the CNN-LSTM, a fixed set of hyperparameters (the best found during the corresponding nested run) was used.

2. **Nested K-Fold:** A 5x3 nested cross-validation. The 5 outer folds were used for performance estimation. The 3 inner folds were used for hyperparameter tuning. For the SVM, the inner loop selected the best number of features. For the CNN-LSTM, the inner loop used Optuna to perform a 25-trial Bayesian optimization search to find the best learning rate, dropout rate, and layer sizes.

The direct implementation of this protocol for the SVM experiments, taken from the project's source code in (`src/cv_strategies.py`), is shown in Listing 3. This function encapsulates the entire nested cross-validation logic. As shown in the code snippet, the outer `for` loop iterates through the performance evaluation folds. Within each of these folds, a `GridSearchCV` object is instantiated. This object encapsulates the entire inner cross-validation loop, searching for the best hyperparameter (`k`) using only the outer training data (`X_train`, `y_train`). The final performance for that fold is then calculated by evaluating the `best_model` on the held-out outer test set (`X_test`). This strict separation ensures that the test data is never used to inform the selection of the best `k`, providing a methodologically sound, unbiased estimate of the pipeline's performance.

```
def run_nested_kfold_cv(X, y, n_splits_outer=5, n_splits_inner=3):
    # Define the outer loop for final performance evaluation.
    outer_cv = StratifiedKFold(n_splits=n_splits_outer, shuffle=True,
    random_state=42)
    # Define the inner loop, which will be used for hyperparameter tuning
    .
    inner_cv = StratifiedKFold(n_splits=n_splits_inner, shuffle=True,
    random_state=42)
```

```
6
7     results = []
8     fold_predictions = []
9
10    # Define the model pipeline to be evaluated.
11    pipeline = Pipeline([
12        ('scaler', StandardScaler()),
13        ('feature_selection', SelectKBest(f_classif)),
14        ('classifier', SVC(kernel='linear', probability=True,
    random_state=42))
15    ])
16
17    # Define the search space for the number of features 'k'.
18    param_grid = {'feature_selection__k': [10, 20, 30, 40, 50]}
19
20    # Iterate through each outer fold for performance evaluation.
21    for train_idx, test_idx in outer_cv.split(X, y):
22        X_train, X_test = X.iloc[train_idx], X.iloc[test_idx]
23        y_train, y_test = y.iloc[train_idx], y.iloc[test_idx]
24
25        # Inner Loop: Use GridSearchCV to find the best 'k'.
26        # This search is performed only on the outer training data.
27        grid_search = GridSearchCV(
28            estimator=pipeline,
29            param_grid=param_grid,
30            cv=inner_cv,
31            scoring='f1_macro'
32        )
33        grid_search.fit(X_train, y_train)
34
35        # The best model found in the inner loop.
36        best_model = grid_search.best_estimator_
37
38        # Evaluate the best model on the held-out outer test set.
39        y_pred = best_model.predict(X_test)
40
41        # (code to store metrics and selected features) ...
42
43    return pd.DataFrame(results), fold_predictions
```

Listing 3: The Python implementation of nested cross-validation for the SVM pipeline, from `src/cv_strategies.py`.

3.5.2 CNN-LSTM Tuning via Bayesian Optimization

The hyperparameter space for the CNN-LSTM is significantly larger and includes continuous values (e.g., learning rate). An exhaustive grid search would be computationally infeasible. Therefore, a more advanced and efficient strategy was employed: Bayesian optimization, implemented using the Optuna library.

This approach iteratively and adaptively explores the parameter space, using the results from previous trials to inform which new combination of hyperparameters is most

likely to yield a better score. For each of the 5 outer folds, a 25-trial Optuna study was conducted. The main evaluation function, shown conceptually in Listing 4, orchestrates this process. It iterates through the outer folds, and for each one, it launches an Optuna study that calls the objective function (shown in Listing 5) to find the best-performing hyperparameters on the inner validation splits.

```
def run_pytorch_nested_cv_with_optuna(sequences_dict, metadata_df,
    n_trials=25):
    # Initialize the outer loop for final performance evaluation
    outer_cv = StratifiedKFold(n_splits=5, ...)

    # (code to align sequence data and labels) ...

    # - Outer Loop
    for fold, (train_val_idx, test_idx) in enumerate(outer_cv.split(
    X_data, y_data)):
        X_train_val, X_test = X_data[train_val_idx], X_data[test_idx]
        y_train_val, y_test = y_data[train_val_idx], y_data[test_idx]

        # - Inner Loop: Hyperparameter Tuning with Optuna
        # Create a new study for each outer fold
        study = optuna.create_study(direction='maximize')

        # Optimize the objective function over n_trials
        study.optimize(
            lambda trial: _objective(trial, X_train_val, y_train_val,
    ...),
            n_trials=n_trials
        )

        best_params = study.best_params

        # - Final Training for this Fold
        # Build the final model with the best found hyperparameters
        final_model = CNNLSTM(**best_params, ...).to(device)

        # Train the final model with early stopping on the full outer
    training data
        best_model = _train_eval_loop(final_model, ...)

        # - Final Evaluation for this Fold
        # Evaluate the best model on the held-out outer test set
        y_true, y_pred, y_prob = _eval_model(best_model, test_loader,
    device)

        # (store final performance metrics) ...
```

Listing 4: Conceptual implementation of nested cross-validation for the CNN-LSTM.

The objective function, shown in Listing 5, defines the search space for key hyperparameters, including learning rate, dropout, layer sizes, and activation function. This approach represents a thorough and computationally efficient method for tuning complex deep learning models within a methodologically sound nested cross-validation framework.

```python
def _objective(trial, X_train_val, y_train_val, device, n_splits_inner):
    """ The objective function that Optuna tries to maximize. """
    # Define the hyperparameter search space for the trial
    params = {
        'learning_rate': trial.suggest_float('learning_rate', 1e-5, 1e-3, log=True),
        'dropout_rate': trial.suggest_float('dropout_rate', 0.2, 0.5),
        'cnn_out_channels': trial.suggest_categorical('cnn_out_channels', [32, 64, 128]),
        'lstm_hidden_dim': trial.suggest_categorical('lstm_hidden_dim', [64, 128]),
        'activation_fn': trial.suggest_categorical('activation_fn', ['silu', 'gelu']),
    }

    # Perform inner cross-validation with these parameters
    inner_cv = StratifiedKFold(n_splits=n_splits_inner, shuffle=True, random_state=42)
    inner_f1_scores = []
    for train_idx, val_idx in inner_cv.split(X_train_val, y_train_val):
        # ... (train and evaluate a model with the trial's params) ...
        inner_f1_scores.append(validation_f1_score)

    # Return the mean score for this trial
    return np.mean(inner_f1_scores)
```

Listing 5: Optuna objective function defining the hyperparameter search space.

3.5.3 Analysis of Optimal Hyperparameters

The nested cross-validation protocol with Bayesian optimization allows for a detailed analysis of the optimal hyperparameters selected for each data type. Table 1 summarizes the average or most frequently selected (mode) values for the key hyperparameters found by the 25-trial Optuna search in each outer fold.

Data Type	learning_rate	# dropout_rate	# cnn_out_channels	# lstm_hidden_dim	activation_fn
Combined	3.21e-04	0.312	64	128	silu
Interview	4.36e-04	0.371	32	128	silu
Reading	1.09e-04	0.382	128	64	silu

Table 1: Summary of the optimal hyperparameters found by the nested Bayesian optimization search for each data type.

The results reveal a non-trivial relationship between the nature of the speech data and the optimal model architecture. A key finding is that the modern `silu` (Swish) activation function was unanimously chosen over `gelu` across all data types, providing strong empirical evidence for its suitability for this speech processing task.

More significantly, the results show a distinct architectural trade-off. For the scripted **Reading** task, the optimizer selected the widest convolutional layers (`cnn_out_channels=128`)

but a shallower recurrent layer (`lstm_hidden_dim=64`). This suggests that for monotonous, read speech, the model benefits most from a powerful local feature extractor to capture fine-grained phonetic detail, while long-range temporal modeling is less critical.

Conversely, for the spontaneous **Interview** task, the optimizer preferred the narrowest convolutional layers (`cnn_out_channels=32`) but a deeper recurrent layer (`lstm_hidden_dim=128`). This indicates that for prosodically rich, dynamic speech, the most important information is contained in the long-term sequential dependencies, which are best captured by a more powerful LSTM, while the local feature extraction can be simpler. The **Combined** model found a logical compromise between these two extremes. This data-driven architectural selection, tailored to the specific characteristics of the input data, is a key advantage of the nested CV protocol and a demonstration of the framework's analytical capabilities.

3.5.4 Evaluation Metrics

The primary metrics for evaluating predictive performance were the macro-averaged **F1-Score** and the **Area Under the Receiver Operating Characteristic Curve (AUC)**, both of which are robust for balanced classification tasks. Feature stability was assessed by counting the frequency with which features were selected in the top-50 most important across the 5 folds of a cross-validation run.

3.6 Scope of Analysis: A Unimodal Acoustic Approach

This project focuses exclusively on the acoustic modality of speech. Textual features from transcripts (NLP) and visual features from video (facial expressions) were deliberately excluded. This unimodal approach was chosen for a critical methodological reason: to isolate and evaluate the predictive power and stability of vocal biomarkers alone. By controlling for other modalities, the experimental framework can provide a clear and unambiguous assessment of how the chosen feature representations and evaluation strategies perform on a purely acoustic basis. The resulting pipelines and trained models from this work are therefore designed to serve as a powerful acoustic component that could be integrated into more complex, multimodal systems in future research.

3.7 Results and Analysis

The complete results from all 24 core experiments are presented and analyzed in this section. The findings are grouped by model type (SVM and CNN-LSTM) and key comparisons. For clarity of visualization F1-Score is presented as the primary graphical metric. The corresponding AUC plots, are available for reference in Section A.1.1.

3.7.1 SVM Baseline Experiments

The initial set of 18 experiments was conducted using the SVM classifier to establish baseline performance and investigate the core research questions. Table 2 provides a comprehensive summary of the performance metrics for all 18 experiments, grouped by data

type. The results reveal that for handcrafted features, the standard k-fold method consistently reports higher scores than the nested approach. Furthermore, models trained on spontaneous Interview data significantly outperform those trained on scripted Reading data, a finding explored in Figure 2.

Performance Metrics Summary for Reading Task Experiments						
Experiment	Mean F1-Score	Std Dev F1-Score	Mean AUC	Std Dev AUC	Mean Accuracy	Std Dev Accuracy
mshds_reading_standard	0.735	0.081	0.810	0.068	0.738	0.076
mshds_reading_nested	0.706	0.176	0.764	0.181	0.711	0.176
opensmile_reading_standard	0.586	0.104	0.572	0.096	0.594	0.100
opensmile_reading_nested	0.563	0.094	0.565	0.105	0.566	0.093
wav2vec2_reading_standard	0.664	0.071	0.703	0.096	0.666	0.070
wav2vec2_reading_nested	0.654	0.036	0.766	0.090	0.658	0.039
Performance Metrics Summary for Interview Task Experiments						
Experiment	Mean F1-Score	Std Dev F1-Score	Mean AUC	Std Dev AUC	Mean Accuracy	Std Dev Accuracy
mshds_interview_standard	0.718	0.101	0.769	0.141	0.724	0.098
mshds_interview_nested	0.709	0.094	0.766	0.121	0.714	0.095
opensmile_interview_standard	0.685	0.040	0.738	0.040	0.688	0.035
opensmile_interview_nested	0.739	0.081	0.798	0.054	0.743	0.077
wav2vec2_interview_standard	0.697	0.120	0.790	0.087	0.699	0.121
wav2vec2_interview_nested	0.687	0.126	0.757	0.087	0.690	0.128
Performance Metrics Summary for Combined Data Experiments						
Experiment	Mean F1-Score	Std Dev F1-Score	Mean AUC	Std Dev AUC	Mean Accuracy	Std Dev Accuracy
mshds_combined_standard	0.758	0.059	0.832	0.106	0.761	0.059
mshds_combined_nested	0.693	0.058	0.783	0.112	0.697	0.060
opensmile_combined_standard	0.676	0.032	0.728	0.033	0.679	0.028
opensmile_combined_nested	0.721	0.107	0.789	0.062	0.726	0.105
wav2vec2_combined_standard	0.732	0.100	0.808	0.049	0.734	0.098
wav2vec2_combined_nested	0.706	0.093	0.806	0.097	0.708	0.092

Table 2: Summary of performance metrics for all 18 SVM experiments, grouped by data type. The heatmap highlights relative performance within each column, yellow indicates the best experiment for a particular evaluation metric and dark purple the worst.

3.7.2 Standard K-Fold Bias

A primary objective was to empirically quantify the optimistic bias of standard k-fold cross-validation. Figure 1 demonstrates this phenomenon using the F1-score. For the MSHDS feature sets, the standard method produced higher scores, indicating a likely optimistic bias as predicted by the literature [7]. Interestingly, for the OpenSMILE set, the bias is negative, suggesting the fixed feature selection of the standard method was suboptimal compared to the adaptive selection in the nested loop. A parallel analysis for the AUC metric, which confirms these findings, is presented in Section A.1 (Figure 10).

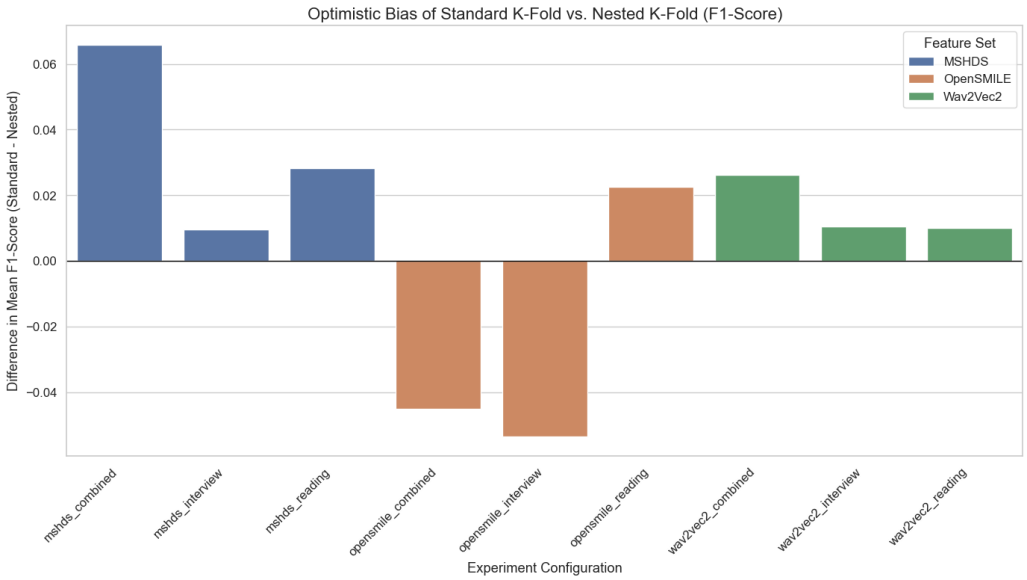

Figure 1: Optimistic bias of Standard vs. Nested K-Fold for SVMs, measured as the difference in Mean F1-Score. Positive bars indicate that the standard method produced a higher, likely biased, performance estimate.

3.7.3 Impact of Speech Elicitation Task

Figure 2 visualizes the performance difference between models trained on spontaneous versus read speech. For the high-dimensional OpenSMILE feature set, switching to spontaneous speech provides a massive performance boost of approximately 17.5% in F1-score, suggesting its comprehensive nature is well-suited to capturing the rich variability of spontaneous speech, a finding that aligns with the work of Braun et al. [23].

Conversely, for the curated MSHDS feature set, the trend is less clear. While the nested method shows a marginal benefit for interview data, the standard k-fold method performs slightly worse. This suggests a potential interaction: the controlled signal of read speech may be sufficient for a small, interpretable feature set when evaluated with a simple method. But its clear for a more complex feature space that rich prosodic and acoustic information in spontaneous speech becomes essential. The corresponding AUC analysis, which supports this conclusion, is available in Section A.1.1 (Figure 11).

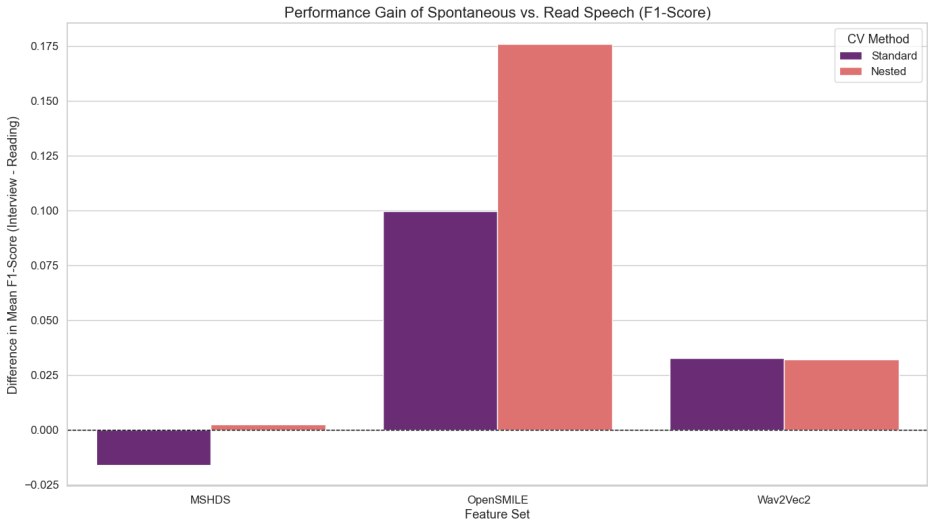

Figure 2: Performance gain of spontaneous (Interview) speech over read (Reading) speech for SVMs (F1-Score).

3.7.4 CNN-LSTM Deep Learning Experiments

The final set of 6 experiments was conducted using the more complex CNN-LSTM architecture on the Wav2Vec2 sequence embeddings to evaluate a state-of-the-art approach. The performance is summarized in Table 3. A key finding is that, unlike the SVMs, the tuned, nested CNN-LSTM consistently and significantly outperforms its standard k-fold counterpart across all data types.

Experiment	Mean F1-Score	Std Dev F1-Score	Mean AUC	Std Dev AUC
wav2vec2_cnn_lstm_standard_reading	0.629	0.134	0.741	0.096
wav2vec2_cnn_lstm_tuned_reading	0.700	0.099	0.779	0.052
wav2vec2_cnn_lstm_standard_interview	0.740	0.088	0.814	0.072
wav2vec2_cnn_lstm_tuned_interview	0.770	0.106	0.865	0.096
wav2vec2_cnn_lstm_standard_combined	0.607	0.157	0.777	0.090
wav2vec2_cnn_lstm_tuned_combined	0.779	0.086	0.847	0.093

Table 3: Final performance summary for all 6 tuned CNN-LSTM experiments.

The training and validation loss curves provide critical insight into the learning process. Figure 3 shows the curves for the final models trained on the Combined dataset. The stable convergence and the clear point where validation loss is minimized demonstrate the effectiveness of the early stopping mechanism. This provides evidence that the adaptive hyperparameter search in the nested protocol leads to a more stable and efficient training process. The full set of loss curves for all data types are provided in Section A.2.1.

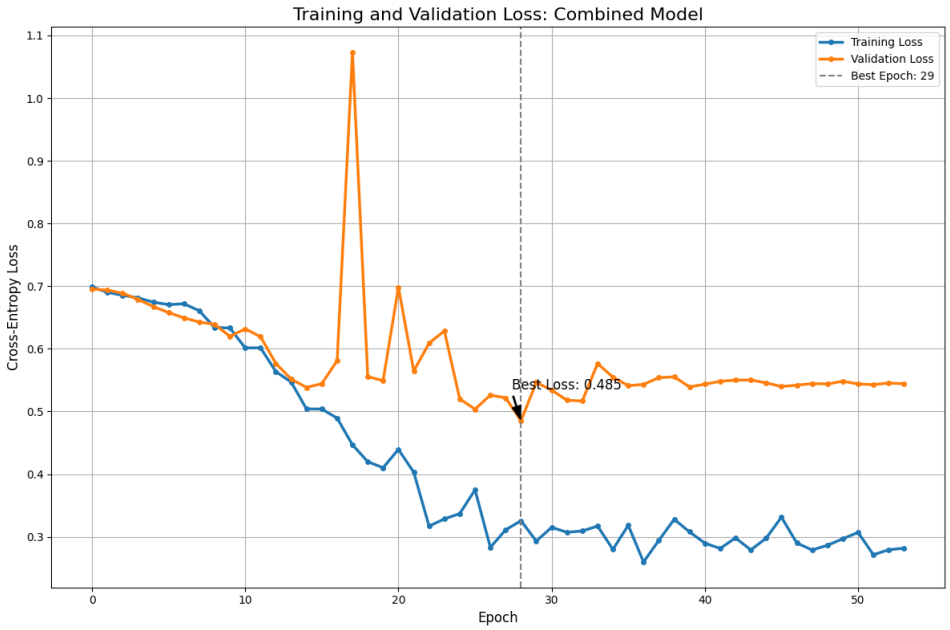

Figure 3: Training and validation loss curve for the final tuned CNN-LSTM model on the Combined dataset.

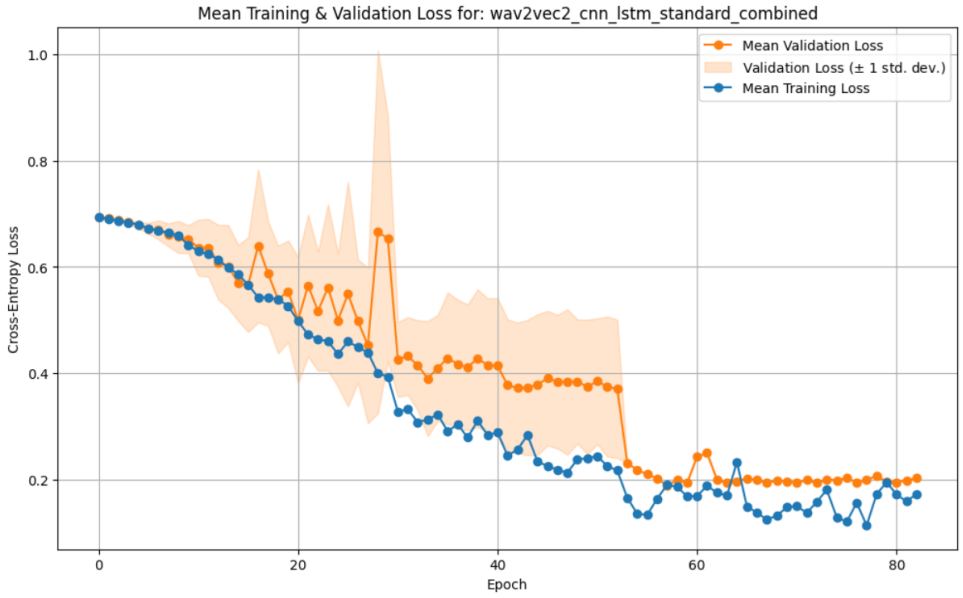

Figure 4: Mean training and validation loss across 5 folds for the Standard K-Fold CNN-LSTM on the Combined dataset.

3.7.5 Feature Stability Analysis

A key component of robustness is the stability of the features selected by the model. Figure 5 shows the most stable features selected by the nested protocol for the mshds feature set on the combined dataset. For the small, curated MSHDS set, the nested approach consistently identifies a stable subset of interpretable biomarkers, with six features being selected in all 5 folds. These include features related to pitch variability (`stdev_F0_Semitone`) and articulatory control (`Mean_Pause_Duration_mean`), which align with clinical descriptions of depressive speech.

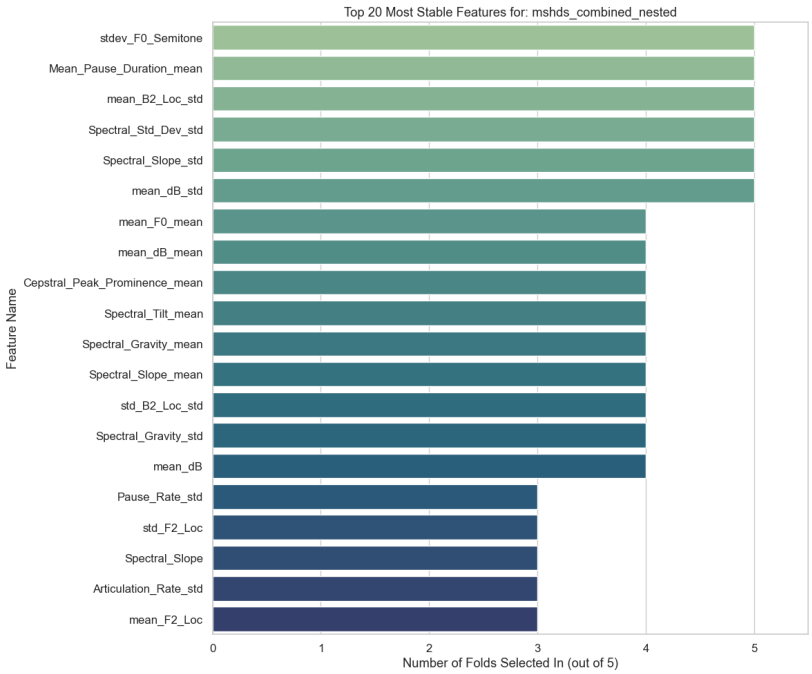

Figure 5: Top 20 most stable features for the MSHDS experiment on the Combined dataset using the robust nested CV protocol.

Surprisingly, according to Figure 6 the nested protocol also identified a highly stable set of features from the high-dimensional OpenSMILE set, with ten features being selected in all 5 folds. This demonstrates the power of the robust evaluation method to find a consistent signal even in a noisy feature space. However, a key trade-off emerges: while more numerous, the top OpenSMILE features (e.g., `pcm_RMSenergy_sma_min_mean`) are acoustically complex and less directly interpretable than their MSHDS counterparts. This highlights a fundamental choice between maximizing the number of stable features and prioritizing their clinical interpretability.

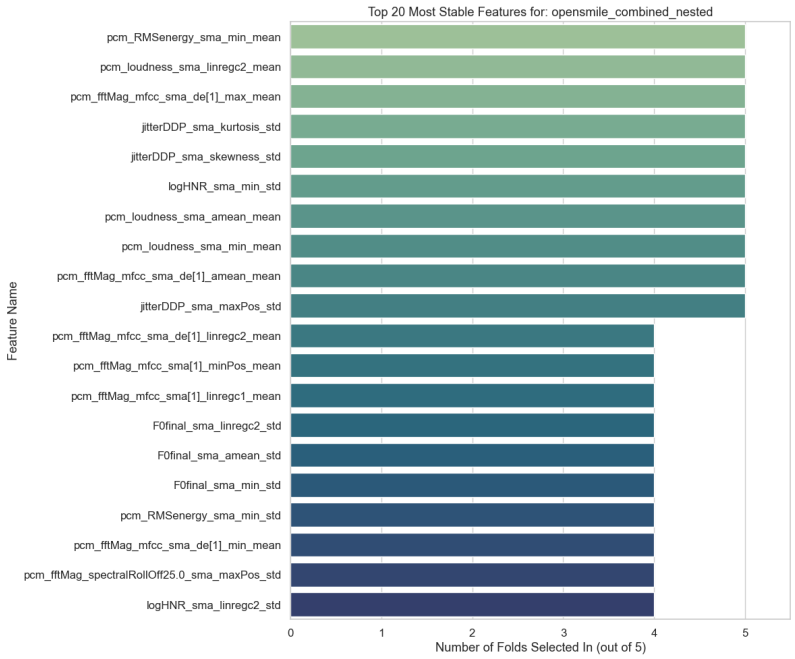

Figure 6: Top 20 most stable features for the OpenSMILE experiment on the Combined dataset using the robust nested CV protocol.

3.7.6 Final Performance and Stability Analysis

The final analysis consolidates the findings to identify the state-of-the-art pipeline developed in this project. The two top-performing configurations were the tuned CNN-LSTM models trained on the Interview and the Combined datasets. A direct comparison of these two models reveals a subtle but important trade-off between different evaluation metrics.

As shown in Table 3, the **Interview model** achieved the highest Mean AUC (0.865), indicating the strongest theoretical discriminative power. However, the **Combined model** achieved a slightly higher Mean F1-Score (0.779 vs. 0.770) and did so with lower variance (Std Dev: 0.086 vs. 0.106). Figure 7 visually compares the ROC curves for these two leading models.

The ROC curves visually confirm that the deep learning architecture, when properly tuned and applied to the rich, sequential data from spontaneous speech, demonstrates superior discriminative power. This is further summarized in the overall trade-off plot (Figure 19).

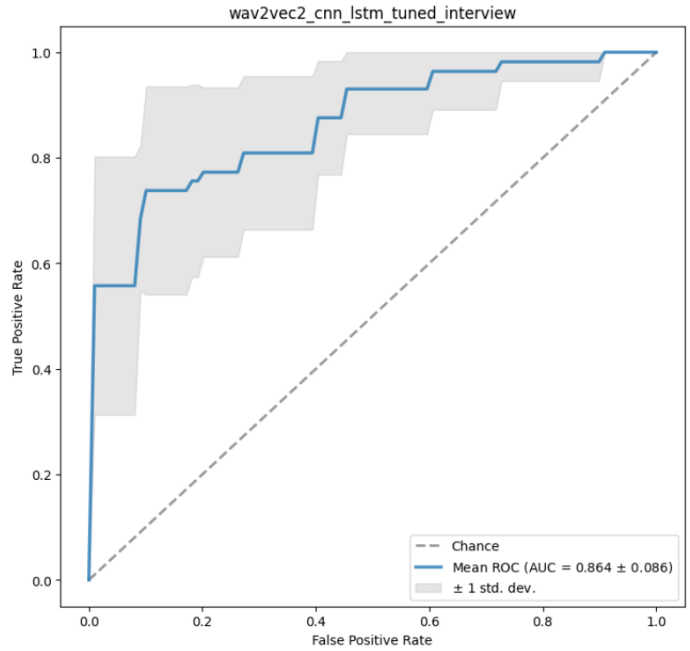

Figure 7: ROC Curve for CNN-LSTM (Wav2Vec2, Interview, Tuned)

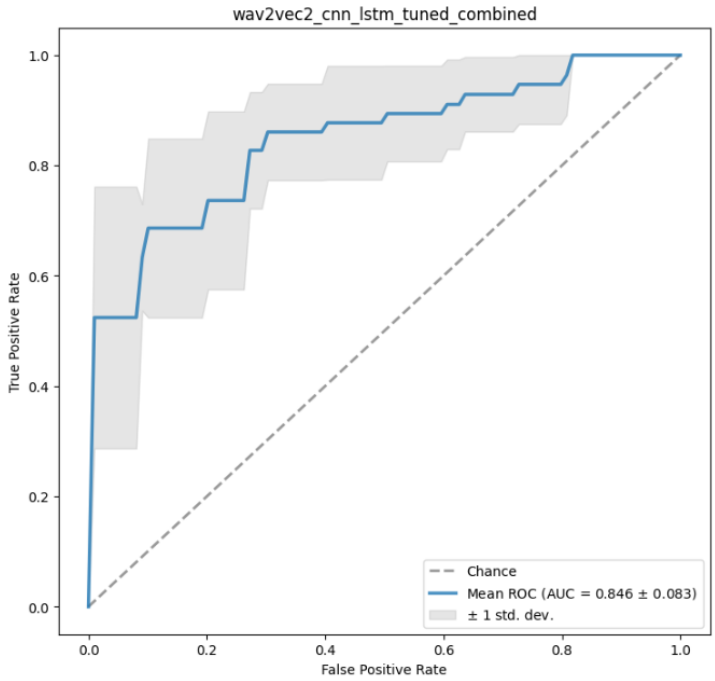

Figure 8: ROC Curve for CNN-LSTM (Wav2Vec2, Combined, Tuned)

While the Interview model's higher AUC is notable, the F1-Score often represents a more practical measure of a classifier's effectiveness, and stability is paramount for clinical reliability. The holistic "Performance vs. Stability" trade-off is summarized in Figure 9. This visualization confirms that while both models are strong performers, the **CNN-LSTM (Tuned) model on the Combined data** occupies the most favorable position in the top-left quadrant. It offers the best balance of high classification performance (F1-Score) and low performance variance, establishing it as the most robust and reliable model developed in this project.

3.7.7 Overall Performance vs. Stability Trade-off

Figure 9 plots mean performance against stability for the Combined dataset, which represents the models trained on all available information. The ideal model is located in the top-left quadrant (high performance, low variance).

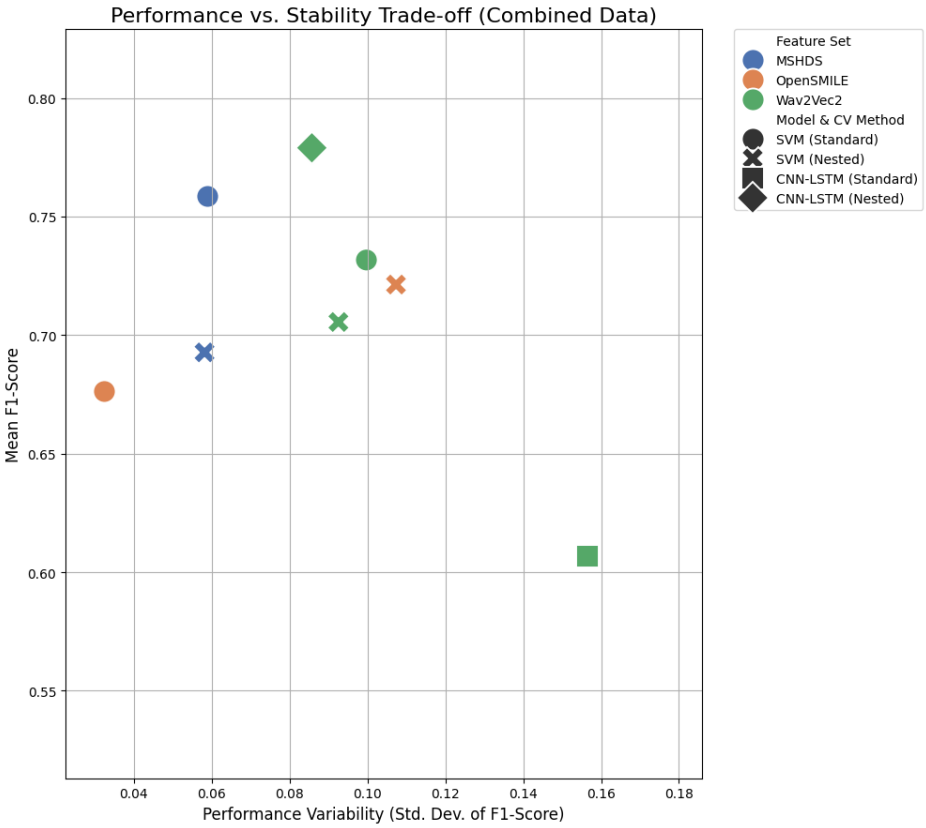

Figure 9: Performance vs. Stability Trade-off for the Combined dataset experiments.

This visualization clearly confirms that the **CNN-LSTM (Nested) model on the Combined data** provides the optimal balance of high performance and reliability. It achieves the highest mean F1-score of ∼0.78 across all experiments with a relatively

low standard deviation F1-score of $\sim$0.09 making it consistent, this establishes it as the state-of-the-art model developed in this project.

The corresponding plots for the individual Reading and Interview tasks, available in Section A.3.1, provide further context. The plot for the Reading data shows generally lower performance across all models. While the Interview data plot reveals high performance for the CNN-LSTM, its stability is lower than that of the Combined model. This suggests that while spontaneous speech is the most information-rich source, the addition of structured read speech provides a regularizing effect that leads to a more stable final model, a key insight from this comparative analysis.

4 Discussion and Conclusion

4.1 Discussion of Findings

The results generated by the experimental framework provide a multi-faceted answer to the core research question and yield several key insights into the robust application of machine learning for speech-based depression detection. This section critically analyzes and interprets these findings in the context of the existing literature, focusing on the impact of evaluation methodology, speech elicitation task, and the trade-offs between different feature and model variants.

4.1.1 The Impact of Cross-Validation on Performance Evaluation

A primary objective of this dissertation was to empirically quantify the optimistic bias of standard k-fold cross-validation when compared to a robust nested protocol. The results from the SVM experiments (Figure 1) clearly demonstrate this phenomenon for hand-crafted feature sets. For both MSHDS and OpenSMILE, the standard k-fold method produced performance estimates that were consistently higher than their nested counterparts. This finding provides real-world evidence on a clinical dataset that supports the theoretical and simulation-based arguments of Ghasemzadeh et al. [7], confirming that feature selection within a simple CV loop leads to inflated and unreliable performance metrics.

Interestingly, for the deep learning models, the effect was reversed: the tuned, nested CNN-LSTM consistently and significantly outperformed the standard k-fold version across all data types. This suggests a more profound issue with the standard approach for complex models. For highly sensitive architectures like neural networks, a fixed set of hyperparameters (used in the standard CV run, which were taken from just one of the nested folds) is highly likely to be suboptimal for the majority of data splits. The adaptive hyperparameter search conducted by the nested protocol is therefore not just a mechanism for producing an unbiased evaluation, but a necessary component for achieving optimal performance in the first place. The argument for the superiority of the nested approach does not rest on having an external test set, but on this empirical evidence combined with its theoretical design to prevent information leakage.

4.1.2 Spontaneous vs. Read Speech: The Value of Context

The comparison between speech elicitation tasks revealed that spontaneous speech is a generally more informative data source, though its benefit interacts with the feature set and evaluation method. The performance gain was most dramatic for the high-dimensional OpenSMILE feature set, suggesting its comprehensive nature is well-suited to capturing the rich prosodic and acoustic variability of spontaneous speech. This aligns with findings from researchers like Alghowinem et al. [24], who note the importance of conversational context in detecting depression. The MSHDS set, however, presented a more nuanced picture where the cleaner signal of read speech sometimes yielded com-

petitive results under the simpler standard k-fold protocol, highlighting the complex trade-offs involved.

The CNN-LSTM results provided an even more nuanced view, highlighting a trade-off between different performance metrics. As shown in Table 3, the **Interview-only model** achieved the highest peak discriminative power (Mean AUC of 0.865). However, the **Combined model** achieved a slightly higher Mean F1-Score (0.779 vs. 0.770) and did so with greater stability (lower performance variance), as seen in the trade-off plot (Figure 9). This suggests that while the raw signal of spontaneous speech is optimal for class separation, the addition of the structured read speech may provide a *regularizing effect*, leading to a more robust final classifier at the default threshold. This interaction between data combination, model architecture, and the choice of evaluation metric is a key insight of this project.

4.1.3 Feature Representation, Model Complexity, and Stability Trade-off

The comparison of the three feature approaches yielded results that highlight the trade-offs between interpretability, performance, and complexity, as summarized in the overall stability trade-off plots (Figure 9), (Figure 19), and (Figure 18). It clearly shows that some models which achieve high mean performance do so at the cost of high variance (instability) across the cross-validation folds. For a clinical application, a model that is slightly less performant but significantly more reliable is often the superior choice. The small, curated MSHDS set proved remarkably effective, especially with the SVM, demonstrating the power of expert-driven feature engineering to create an interpretable and high-performing baseline. The learned Wav2Vec2 embeddings, when paired with the CNN-LSTM, ultimately achieved the highest robustly-evaluated performance, confirming the strength of modern, deep learning-based representations as noted by Campbell et al. [18].

The feature stability analysis (Figure 5) provided further insight. The nested CV approach consistently identified a smaller and more stable set of important features. For the MSHDS set, features related to pitch variability (`stdev_F0_Semitone`), voice quality (`Cepstral_Peak_Prominence`), and articulatory control (`Mean_Pause_Duration_mean`) were found to be the most reliable biomarkers, aligning with clinical descriptions of depressive speech. This demonstrates the framework's ability to identify trustworthy biomarkers.

In analysis of the optimal hyperparameters found during the nested CV (Table 1) provided a deeper insight. It revealed that the ideal model architecture is highly dependent on the nature of the input data. For scripted, monotonous read speech, a wider CNN feature extractor was preferred, while for dynamic, spontaneous speech, a deeper LSTM layer was favored. This data-driven architectural adaptation, automatically discovered by the evaluation framework, is a key finding, demonstrating that a "one-size-fits-all" model is likely suboptimal.

4.2 Limitations and Future Work

While this project provides a comprehensive methodological analysis, several limitations should be acknowledged. The entire study was conducted on a single, albeit high-quality, corpus of Italian speakers. The findings regarding specific feature importance and performance metrics may not generalize directly to other languages, cultures, or datasets with different recording protocols without further investigation. However, the methodological principles demonstrated the importance of nested CV, the value of spontaneous speech, and the analysis of model-feature interaction are language-agnostic and provide a valuable blueprint for future studies. A key avenue for future work would be to apply this framework to other corpora, such as the English-language DAIC-WOZ, to investigate the cross-lingual robustness of these findings.

Furthermore, the scope of the classifier comparison was focused on the most natural pairings of features and models. The CNN-LSTM architecture was exclusively evaluated with the sequential Wav2Vec2 embeddings, as these models are designed to process temporal data. The handcrafted feature sets, MSHDS and OpenSMILE, were evaluated using SVMs on session-level summary statistics. Applying these handcrafted feature approaches to the CNN-LSTM would have required a separate, non-trivial feature extraction process to generate frame-by-frame Low-Level Descriptors (LLDs) rather than summary functionals. While the direct comparison of Wav2Vec2 across both SVM and CNN-LSTM provided a clear view of the value of temporal modeling, a valuable extension for future work would be to generate these LLDs for the handcrafted sets to enable a complete, all-to-all comparison of feature sets and model architectures.

It should also be said the hyperparameter search for the CNN-LSTM, while implemented using nested cross-validation and Bayesian optimization, was constrained by computational resources. The search space for model architecture parameters, such as the number of CNN output channels and LSTM hidden dimensions, was limited to a maximum of 128. While the available GPU (NVIDIA RTX 3090) is powerful, the memory requirements of processing long, variable-length sequences prevented the exploration of even larger architectures. Similarly, the number of Optuna trials was set to 25. While this represents a thorough search, a larger number of trials could potentially discover a more optimal set of hyperparameters. This highlights a real-world computational bottleneck in deep learning research: the trade-off between the exhaustiveness of the search and the feasibility of the experiment.

Finally, the data augmentation experiments, which were initially planned, proved challenging, with initial results showing a significant degradation in performance. This suggests that augmentation strategies must be carefully designed and tuned to the specific feature representation being used, as powerful learned embeddings like Wav2Vec2 may be sensitive to transformations that distort the subtle acoustic patterns they have learned to rely on. A systematic investigation into optimal augmentation for learned speech embeddings would be a valuable area for future research. Finally, while the CNN-LSTM architecture is powerful, exploring other advanced architectures, such as attention-based Transformers applied directly to the audio, could yield further performance improvements.

4.3 Conclusion

This dissertation successfully developed and applied a robust framework for evaluating machine learning pipelines for speech-based depression detection. The experimental results provide strong empirical evidence for several key conclusions. First, robust evaluation methods like nested cross-validation are critical for obtaining trustworthy performance estimates and, for complex models, for achieving optimal performance. Second, spontaneous speech is a significantly more informative data source than read speech for this task, particularly for high-dimensional and learned feature sets. Third, while curated handcrafted features provide a strong and interpretable baseline, the combination of foundational model embeddings (Wav2Vec2) with a matched deep learning architecture (CNN-LSTM) yields the highest and most robust performance when considering the trade-off between classification metrics (F1-Score) and stability.

The primary contribution of this work is not the development of a single state-of-the-art classifier, but rather the rigorous, comparative analysis that emphasises the importance of methodological choices. By demonstrating the complex interactions between feature sets, data types, and evaluation strategies, this project provides a clear set of findings and a methodological blueprint that can help guide the field toward developing more reliable and clinically translatable speech-based health assessment technologies. The final trained CNN-LSTM model, saved as a project artefact, represents a tangible outcome of this design and evaluation process.

5 Legal, Social, Ethical and Professional Issues

The development and deployment of machine learning models for healthcare applications, particularly in a sensitive area like mental health, carries significant legal, social, ethical, and professional responsibilities. This project was conducted with a keen awareness of these issues, adhering to the principles outlined by professional bodies such as the British Computer Society (BCS) and the Institution of Engineering and Technology (IET).

5.1 Ethical Considerations and Data Privacy

The primary ethical concern in this project relates to the use of human participant data containing sensitive health information. The Androids Corpus, while publicly available for research, contains voice recordings that are inherently identifiable personal data. The ethical responsibility is to ensure this data is handled securely and used solely for the stated research purpose. To uphold this, all data was stored locally on an encrypted drive, and no attempt was made to de-anonymize participants. Furthermore, all findings are reported only in aggregate, ensuring no individual's data can be singled out.

This approach aligns with legal frameworks such as the General Data Protection Regulation (GDPR) and the ethical principles of the BCS's Code of Conduct regarding public interest and privacy. A deeper ethical consideration is that the model itself infers new, highly sensitive information, a potential health diagnosis, from the voice data. The responsible handling of this inferred data is paramount, and the project's focus on privacy and aggregate reporting respects the autonomy and dignity of the data subjects.

5.2 Software Trustworthiness and Reproducibility

A core theme of this dissertation is the trustworthiness of machine learning models. The professional aspect of this is ensuring that the research itself is transparent, accountable, and reproducible. To this end, several professional standards were maintained:

- **Version Control:** The entire project codebase was managed using Git, providing a complete and auditable history of the development process.
- **Code Modularity:** The code was structured into distinct, reusable modules for data loading, feature extraction, and evaluation, adhering to software engineering best practices for clarity and maintainability.
- **Dependency Management:** A Conda environment was used to create a self-contained, reproducible software environment, with all dependencies explicitly managed to guarantee that others can replicate the results precisely.
- **Methodological Transparency:** The dissertation provides a detailed description of all methodological choices, from feature extraction parameters to the exact cross-validation strategies employed, allowing for full scrutiny and replication by other researchers.

This commitment to reproducibility directly demonstrates the principle of 'Professional Competence and Integrity' as outlined by the BCS. The technical investigation into the optimistic bias of standard k-fold cross-validation is itself a contribution to the discussion on software trustworthiness in AI. It provides empirical evidence for why more robust methods are necessary to produce reliable claims, which is essential for maintaining public trust in the profession.

5.3 Intellectual Property and Open Science

This project was built upon the principles of open science. It leverages a publicly available dataset and exclusively uses open-source software libraries. This approach ensures that the work is grounded in the collective knowledge of the research community and that the methods can be adopted and extended by others without proprietary restrictions. This commitment fulfills the professional responsibility to acknowledge the work of others through proper citation and contributes to the advancement of the field. By making the methodology transparent, it promotes an equitable research environment, lowering the barrier for other researchers to build upon this work, thereby fostering sustainable scientific progress.

5.4 Thoughtful Discussion of Project Impact

5.4.1 Public Well-being and Social Implications

The successful development of this technology could have a significant positive impact on public well-being. A reliable tool for remote health assessment could increase access to mental healthcare, particularly for those in underserved or remote communities. It could also enable more proactive interventions by detecting signs of relapse early, potentially reducing the long-term burden of Major Depressive Disorder on individuals and society.

However, there are considerable social and ethical risks that must be addressed. Algorithmic bias is a primary concern. A model trained on a specific demographic may not perform well for individuals with different accents, languages, ages, or genders. This could lead to a 'digital health divide', where the technology benefits some groups while failing or even misdiagnosing others, potentially widening existing health disparities. Accessibility is another key social factor; the system must be usable by individuals with disabilities, including those with speech impediments that are not related to depression.

5.4.2 Economic and Commercial Factors

From an economic perspective, this technology offers the potential for significant cost savings in public and private healthcare systems. By enabling remote monitoring, it could reduce the need for frequent and costly clinical visits, freeing up specialist resources. For commercialization, a viable product would need a clear go-to-market strategy, likely involving partnerships with telehealth platforms, healthcare providers, or insurance companies.

The commercial pathway is accompanied by major legal and regulatory challenges. In Europe, such a tool would likely be classified as a medical device and require certification under the Medical Device Regulation (MDR). This legal process demands rigorous, large-scale clinical trials to prove safety and efficacy, far exceeding the scope of this academic project. Furthermore, commercial deployment would necessitate clear legal frameworks for liability. If the model produces an incorrect assessment, determining responsibility between the software developer, the clinician, and the healthcare provider is a complex legal question that remains a significant hurdle for AI in medicine.

5.4.3 Sustainability and Future Responsibility

The long-term sustainability of this technology depends on more than just technical performance; it depends on public and clinical trust. The focus of this dissertation on methodological rigor, unbiased evaluation, and reproducibility is a direct attempt to build that trust. By demonstrating the pitfalls of simplistic evaluation methods, this project advocates for a standard of scientific diligence that is essential for the sustainable development of the field. While the environmental impact of this specific project is minimal, the broader trend of using large, pre-trained models carries a significant computational and energy cost. Future work must consider the environmental sustainability of these increasingly complex architectures. Ultimately, this work is a response to the professional imperative to build AI systems that are not just powerful, but are also safe, fair, and trustworthy, thereby contributing to a sustainable and beneficial role for AI in healthcare.

References

[1] R. Kohli and S. Tan, "The role of digital health in the future of healthcare," *Academic Medicine*, vol. 92, no. 1, pp. 31–35, 2017.

[2] N. Cummins, B. Schuller, and J. Krajewski, "A review of the use of modern speech recognition systems in health care," *Health and Technology*, vol. 5, pp. 179–187, 2015.

[3] G. . M. D. Collaborators, "Global, regional, and national burden of 12 mental disorders in 204 countries and territories, 1990–2019: a systematic analysis for the global burden of disease study 2019," *The Lancet Psychiatry*, vol. 9, no. 2, pp. 137–150, 2022.

[4] F. Matcham, C. Barattieri di San Pietro, V. Bulgari, G. de Girolamo, R. Dobson, H. Eriksson, A. Folarin, J. Haro, M. Kerz, F. Lamers, *et al.*, "Remote assessment of disease and relapse in major depressive disorder (radar-mdd): a multi-centre prospective cohort study protocol," *BMC psychiatry*, vol. 19, no. 1, pp. 1–18, 2019.

[5] J. Dineley, N. Cummins, *et al.*, "Responsible development and translation of clinical speech analytics," in *ISCA INTERSPEECH*, 2024. Tutorial.

[6] J. L. Pierce, K. Tanner, R. M. Merrill, L. Shnowske, and N. Roy, "Acoustic variability in the healthy female voice: Within-day and across-days effects and influencing factors," *Journal of Voice*, vol. 35, no. 4, pp. 653–e1, 2021.

[7] H. Ghasemzadeh, R. E. Hillman, and D. D. Mehta, "Toward generalizable machine learning models in speech, language, and hearing sciences: Estimating sample size and reducing overfitting," *Journal of Speech, Language, and Hearing Research*, vol. 67, no. 3, pp. 753–781, 2024.

[8] J. Torous, J. Firth, and J. Torous, "New tools for new research in psychiatry: a scalable and customizable platform to empower data driven smartphone research," *JMIR mental health*, vol. 3, no. 2, p. e5165, 2016.

[9] L. M. Babrak, G. Seda, K. Kask, *et al.*, "How can digital biomarkers be integrated in the clinical research of neurodegenerative diseases?," *Expert Review of Neurotherapeutics*, vol. 19, no. 8, pp. 711–720, 2019.

[10] A. Tsanas, M. A. Little, P. E. McSharry, and L. O. Ramig, "Novel speech signal processing algorithms for high-accuracy classification of parkinson's disease," *IEEE transactions on biomedical engineering*, vol. 59, no. 5, pp. 1264–1271, 2012.

[11] M. Anselmi, J. Caligagan, L. Zinman, and Y. Yunusova, "Acoustic analysis of speech in early-and late-onset amyotrophic lateral sclerosis," *Journal of Speech, Language, and Hearing Research*, vol. 65, no. 5, pp. 1760–1771, 2022.

[12] D. Low, K. Bentley, and S. Ghosh, "Automated assessment ofpsychiatric disorders using speech: A systematic review," *Laryngoscope Investigative Otolaryngology*, vol. 5, no. 1, pp. 96–116, 2020.

[13] F. Eyben, M. Wöllmer, and B. Schuller, "Opensmile: the munich versatile and fast open-source audio feature extractor," in *Proceedings of the 18th ACM international conference on Multimedia*, pp. 1459–1462, 2010.

[14] B. Schuller, S. Steidl, A. Batliner, J. Hirschberg, J. K. Burgoon, A. Baird, A. Elkins, Y. Zhang, E. Coutinho, and K. Evanini, "The interspeech 2016 computational paralinguistics challenge: Deception, sincerity & native language," in *Interspeech 2016*, 2016.

[15] Y. Maryn, P. Corthals, P. Van Cauwenberge, M. De Bodt, and F. L. Wuyts, "The cepstral peak prominence: a reliable and valid measure in dysphonia diagnostics," *Journal of Voice*, vol. 23, no. 3, pp. 323–334, 2009.

[16] Y. Jadoul, B. Thompson, and B. De Boer, "Parselmouth: A python library for the praat phonetics software," *Journal of Phonetics*, vol. 71, pp. 1–15, 2018.

[17] A. Baevski, Y. Zhou, A.-r. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," *arXiv preprint arXiv:2006.11477*, 2020.

[18] A. G. Campbell, J. Dineley, F. Matcham, R. J. Dobson, and N. Cummins, "Classifying depression symptom severity using the radar-mdd longitudinal speech corpus: Are personalised models and self-supervised learning features the future of remote monitoring?," *arXiv preprint arXiv:2306.00768*, 2023.

[19] G. C. Cawley and N. L. Talbot, "On over-fitting in model selection and subsequent selection bias in performance evaluation," *Journal of Machine Learning Research*, vol. 11, no. Jul, pp. 2079–2107, 2010.

[20] A. Vabalas, E. Gowen, E. Poliakoff, and A. J. Casson, "Machine learning algorithm validation with a limited sample size," *PloS one*, vol. 14, no. 11, p. e0224365, 2019.

[21] A. Kalousis, J. Prados, and M. Hilario, "Stability of feature selection algorithms: a study on high-dimensional spaces," *Knowledge and information systems*, vol. 12, pp. 95–116, 2007.

[22] A. Batliner, R. Huber, J. Spilker, and E. Nöth, "You are not alone!: children interacting with an emotional conversational agent," in *Speech Prosody 2006*, pp. 325–328, 2006.

[23] S. Braun, S. Gpp, G. An, N. Cummins, and B. W. Schuller, "Comparing read and spontaneous speech for the detection of clinical depression," *IEEE journal of selected topics in signal processing*, vol. 14, no. 2, pp. 374–382, 2020.

[24] S. Alghowinem, R. Goecke, J. Epps, M. Wagner, and J. F. Cohn, "Detecting depression: a comparison between spontaneous and read speech," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5115–5119, IEEE, 2016.

[25] C. Clavel, A. García-Pablos, J. Gómez-García, B. García-Zapirain, S. Perez-Roche, I. García-Lekue, N. Cummins, and B. W. Schuller, "The androids corpus: A new publicly available challenging corpus for speech-based depression detection," *IEEE Transactions on Affective Computing*, 2023.

A Appendix

A.1 Supplementary SVM Figures

A.1.1 Bias and Performance Gain

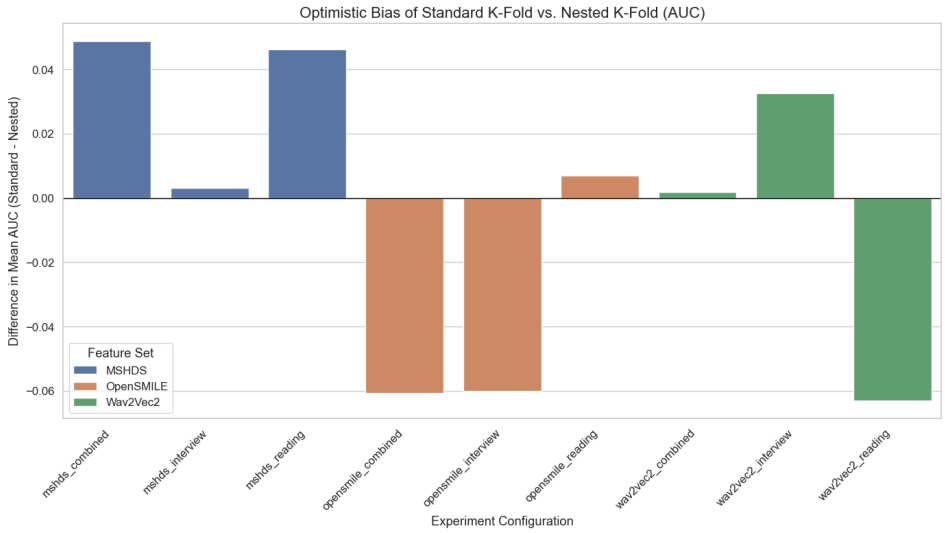

Figure 10: Optimistic bias of Standard vs. Nested K-Fold for SVMs (AUC).

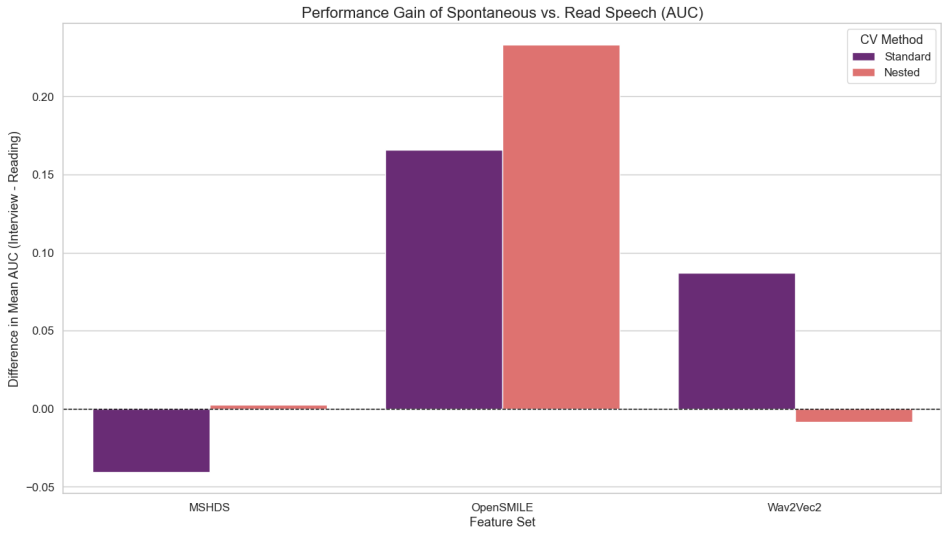

Figure 11: Performance gain of spontaneous (Interview) speech over read (Reading) speech for SVMs (AUC).

A.1.2 SVM ROC Curves

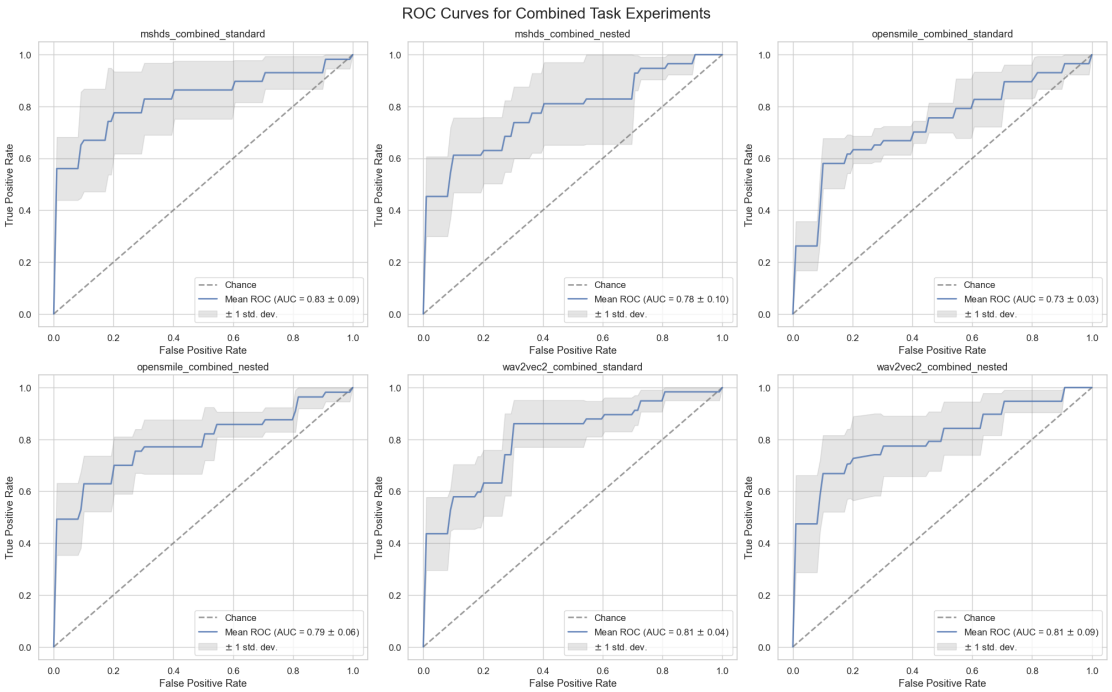

Figure 12: Full ROC Curves for all SVM experiments on the Combined dataset.

A.2 Supplementary CNN-LSTM Figures

A.2.1 All CNN-LSTM Loss Curves

This section contains the full set of training and validation loss curves for the final trained CNN-LSTM models across the Reading and Interview data types.

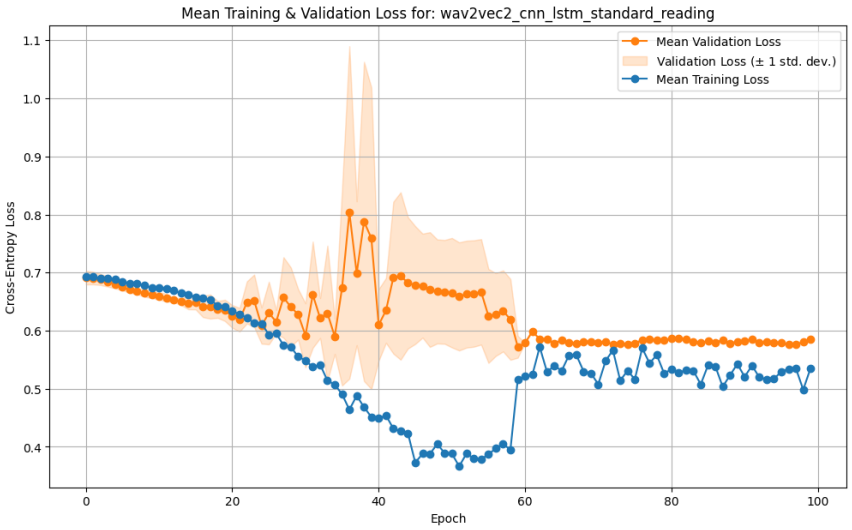

Figure 13: Mean training and validation loss for the Standard K-Fold CNN-LSTM on the Reading Task dataset.

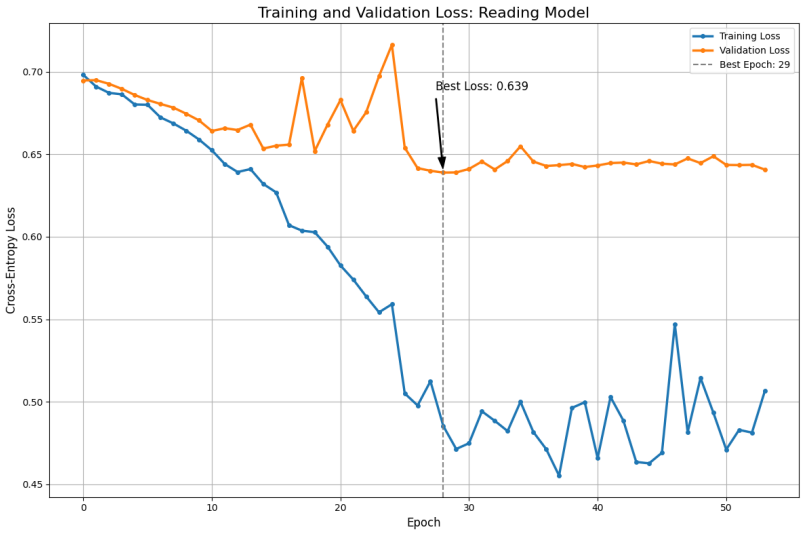

Figure 14: Training and validation loss for the final tuned CNN-LSTM model on the Reading Task dataset.

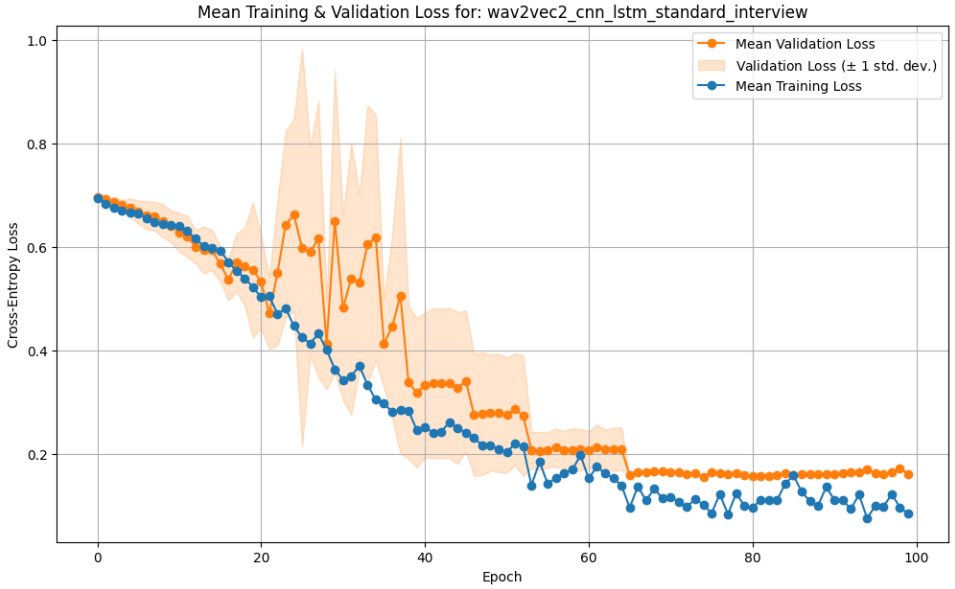

Figure 15: Mean training and validation loss for the Standard K-Fold CNN-LSTM on the Interview Task dataset.

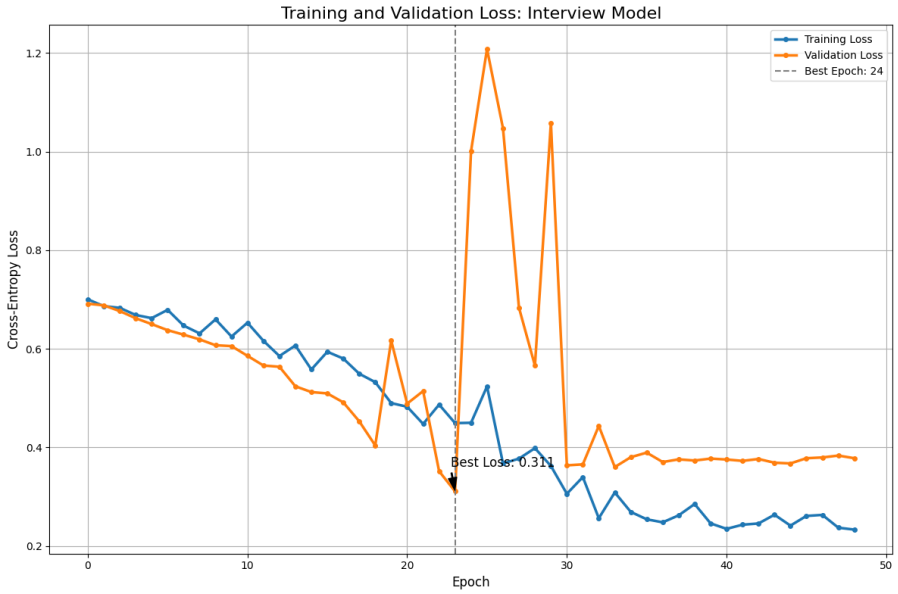

Figure 16: Training and validation loss for the final tuned CNN-LSTM model on the Interview Task dataset.

A.2.2 CNN-LSTM ROC Curves

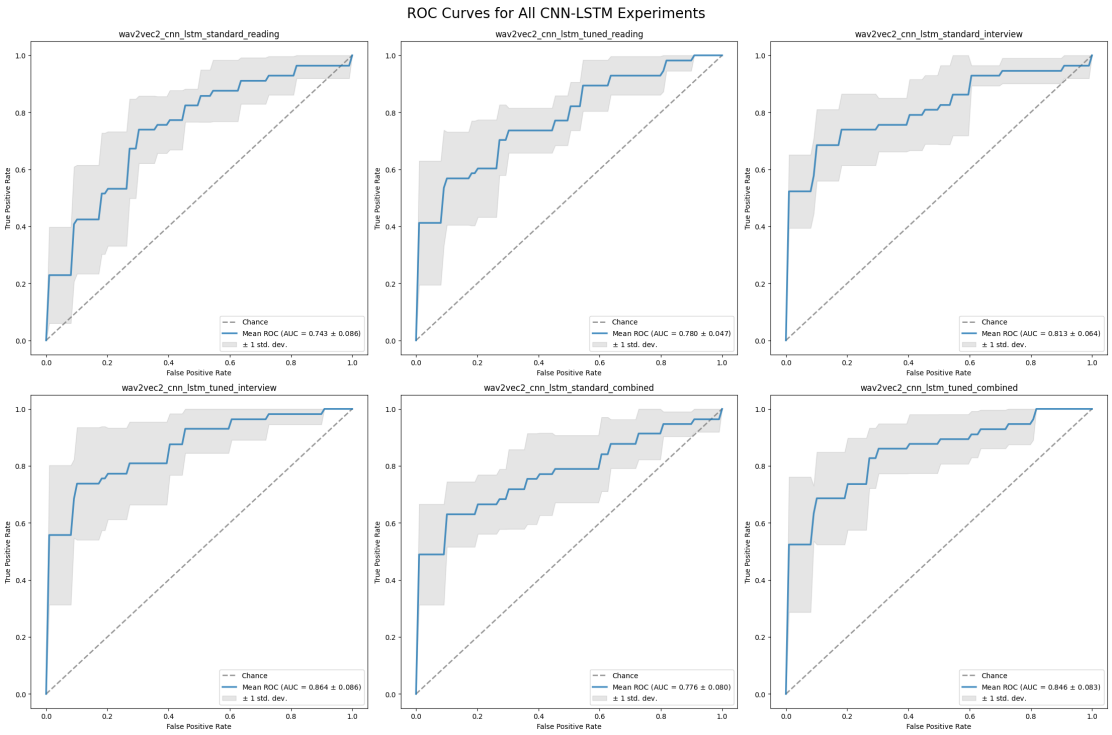

Figure 17: Full ROC Curves for all 6 CNN-LSTM experiments.

A.3 All Experiments Figures

A.3.1 Performance vs. Stability Plots

The following figures show the performance vs. stability trade-off for the Reading and Interview datasets individually, providing context for the main Combined data plot presented in Figure 9.

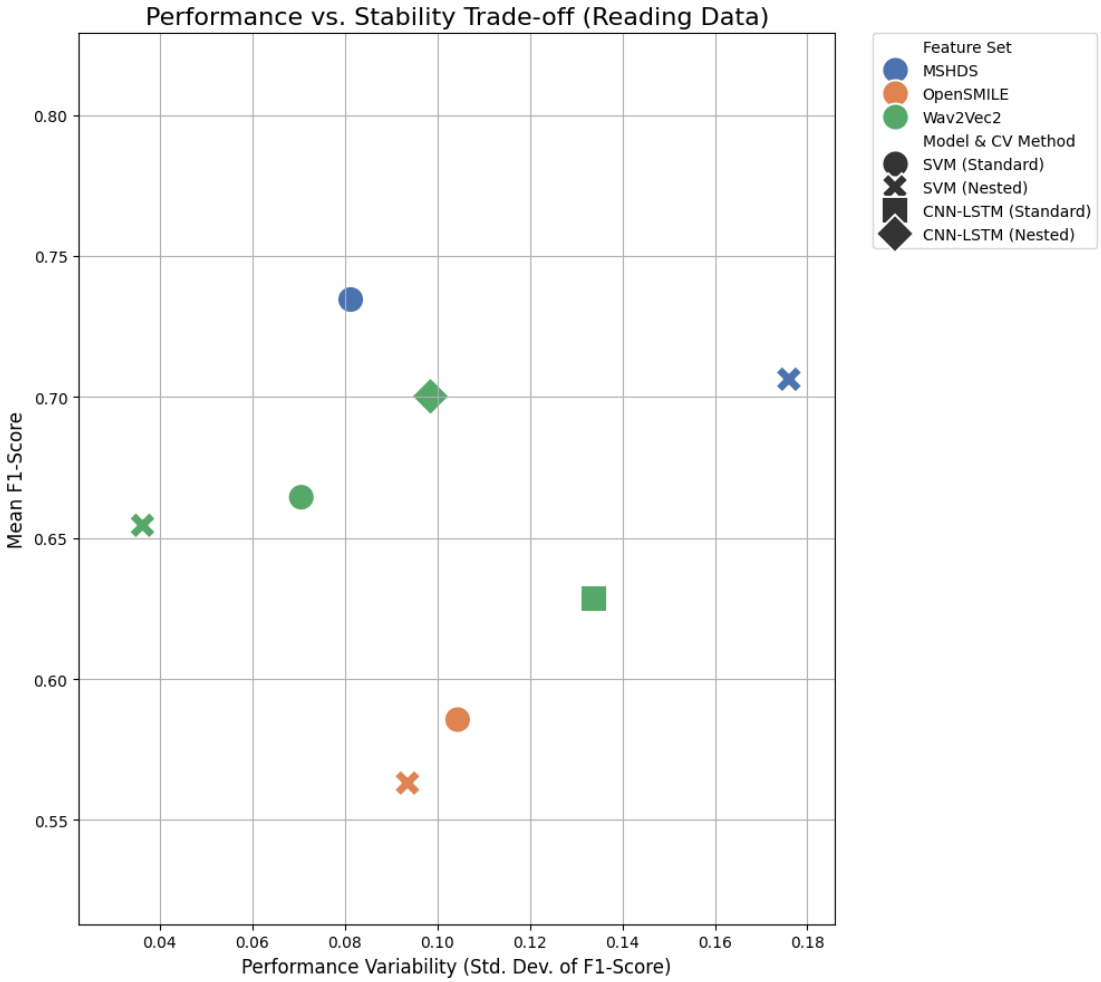

Figure 18: Performance vs. Stability Trade-off for the Reading dataset experiments.

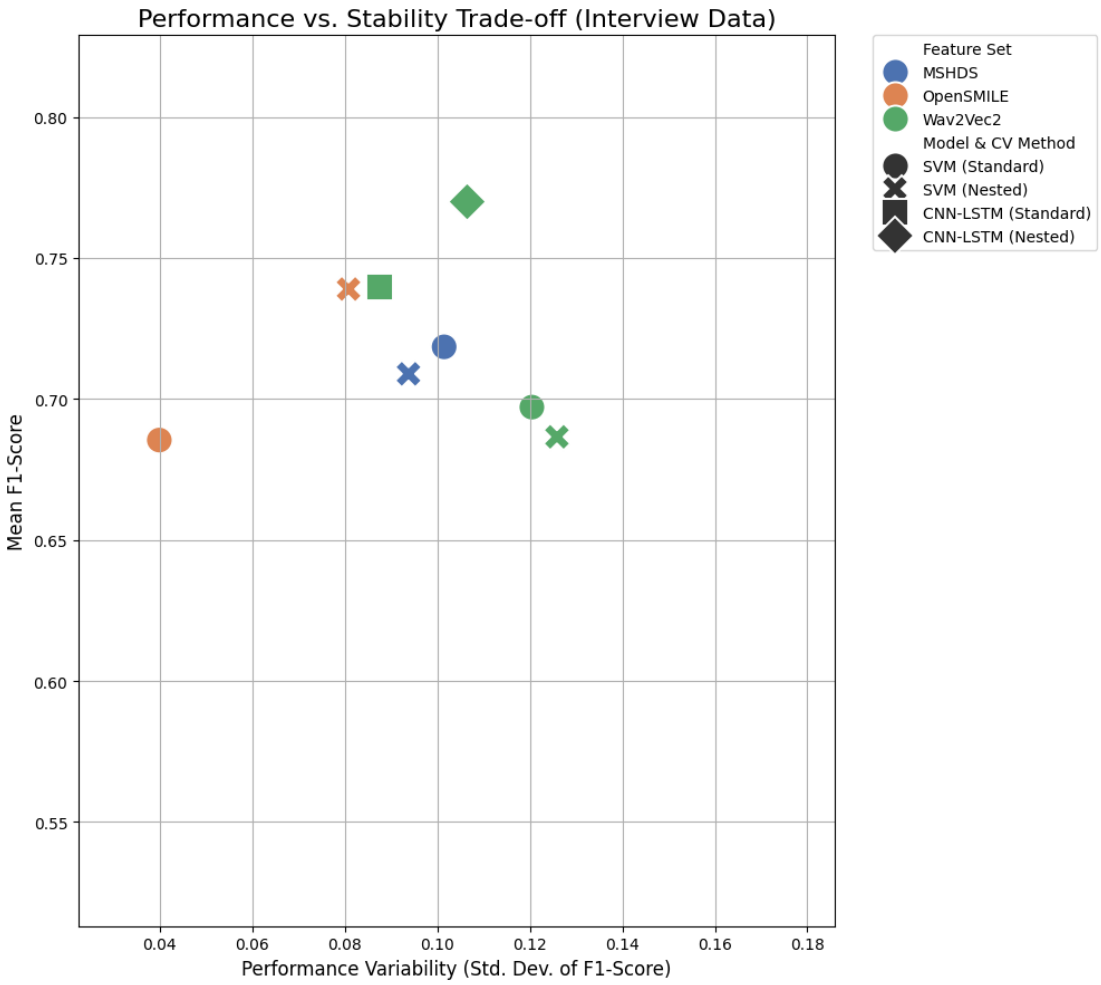

Figure 19: Performance vs. Stability Trade-off for the Interview dataset experiments.